Please amend the present application as follows:

Claims

The following is a copy of Applicant's claims that identifies language being added with underlining ("___") and language being deleted with strikethrough ("—"), as is applicable:

1-9. (Canceled)

10. (Currently amended) A method for controlling the execution of a child process created from a parent process, ~~wherein the parent process is instrumented by a software tool,~~ the method comprising:

instrumenting a parent process;

monitoring execution of the parent process with a process monitor to collect information as to the run-time behavior of the parent process;

before a vfork system call is executed, receiving with the process monitor indicia from the parent process that a vfork system call will be executed by the parent process;

suspending execution of the parent process;

extracting with the process monitor a process identifier from the indicia ~~of the vfork system call~~, the process identifier ~~corresponding to~~ identifying a child process to be generated by the parent process when the parent process executes the vfork system call;

setting with the process monitor a process monitor thread to observe trace events generated by the child process; ~~and~~

resuming execution of the parent process to enable the parent process to execute the vfork system call; and

again suspending execution of the parent process.

11. (Currently amended) The method of claim 10, further comprising:

waiting for indicia that the child process has invoked at least one of an exec system call and an _exit system call or has been terminated by an operating system; ~~and~~

setting a process monitor thread to observe trace events generated by the parent process; and

resuming execution of the parent process.

12. (Canceled)

13. (Currently amended) The method of claim 10, wherein receiving indicia comprises receiving a pre-fork event that includes a process identifier that identifies the child process.

14-28. (Canceled)

29. (Currently amended) A method for controllably switching a target process of a process monitor thread between an instrumented parent process and a child process generated by the parent process, the method comprising:

checking whether the successful initiation of the child process can be asserted;

when the successful initiation of the child process cannot be asserted, checking if the parent process responsible for creating the child process received indicia of a failure of a vfork system call designated to create the child process by searching for a trace event while performing a non-blocking trace wait on the parent process;

when the indicia has not been received,

waiting an amount of time before rechecking for the successful initiation of the child process; otherwise,

notifying a software monitor of the unsuccessful initiation of the child process and resuming execution of the parent process;

monitoring the parent process;

otherwise, when the successful initiation of the child process can be asserted,

monitoring the successfully created child process.

30. (Previously presented) The method of claim 29, wherein the step of checking whether the successful initiation of the child process can be asserted comprises verifying the success of a trace event by using the process identifier of the child process.

31. (Canceled)

32. (Previously presented) The method of claim 29, further comprising:

aborting child process monitoring when the initiation of the child process is unsuccessful.

33-34. (Canceled)

35. (Currently amended) A computer readable ~~medium~~ memory that stores a system, the system comprising:

~~logic responsive to a pre-fork event, the pre-fork event responsive to a vfork system call wherein the pre-fork event includes indicia that identifies a child process to be created in accordance with the vfork system call;~~

a parent process configured to, before a vfork call is executed by the parent process, generate a pre-fork event that contains a process identifier of a child process that will be spawned from the parent process when the vfork system call is executed by the parent process; and

a process monitor configured to receive the pre-fork event and process identifier before the vfork system call is executed by the parent process, suspend execution of the parent process, and generate a process monitor thread that enables observation of trace events generated by the child process.

36. (Canceled).

37. (New) The method of claim 1, further comprising analyzing run-time data observed during execution of the parent process once it is determined that the parent process has generated at least one of an exec system call and an _exit system call or has been terminated by an operating system.

38. (New) The computer readable memory of claim 35, wherein the process monitor is further configured to generate a process monitor that enables observation of the parent process upon resumption of execution of the parent process after receipt of indicia that the child process has invoked at least one of an exec system call and an _exit system call or has been terminated by an operating system.

39. (New) The computer readable memory of claim 38, wherein the process monitor is further configured to resume execution of the parent process.